PP64

PROM PROGRAMMER

FÜR

COMMODORE

C64 – SX64 – C 128

Merlin Data Elektronik
Kay-Römerfeld 14, D-8261 Tittmoning, Tel. 08683/933

Inhaltsverzeichnis:

Inhalt	Seite
Allgemeines	1
Inbetriebnahme	1
Besonderheiten des 27512 EPROM	3
PP64 MENÜ - Programm	3
PROM lesen	4
PROM programmieren	4
von Disk laden	5
auf Disk speichern	5
Adressen ändern	5
PROM-Typ ändern	6
Diskverzeichnis	6
Diskkommandos	6
Blankcheck	7
Vergleichen	7
Modulgenerator	7
PP64 ERWEITERUNG	8
Direktmodus Befehle	8
BASIC Befehle	9
Tabelle Kontroll- & Programmierwort	9
Format von EPROM Dateien	11
Programmierungsarten	11
PP64 und SX-64	13
Maschinensprache Monitor	13
Belegung der EPROM Typen	21

MERLIN DATA ELEKTRONIK
PP64 Modul

Allgemeines:
Dies ist die neueste Version des bekannten PP64 EPROM-Programmers für den C=64. Da die Zeit nicht stehenbleibt – in der Elektronik schon gar nicht – sind einige neue EPROM- und EEPROM-Typen auf den Markt gekommen. Um unseren Kunden den besten Service zu bieten passen wir die Software des PP64 immer an diese neuen Typen an.

Durch die Vielzahl von Typen wurde die Programmierung immer unübersichtlicher. Aus diesem Grunde wurde das Programm PP64 Menü entwickelt, welches von Disk geladen wurde. Einige Punkte dieses Programms waren aber verbesserungswürdig, weshalb wir uns entschlossen eine komplett neue Version zu schreiben und diese in ein Modul zu geben.

Die Verbesserungen betreffen:

- 32 kByte RAM frei für Daten
- 4 Programmierspannungen 5, 12.5, 21 und 25 V
- EEPROMs bis 256 kBit (X 2864 A) und X 28256 A)
- EPROMs bis 512 kBit (I 27512)
- 3 intelligente Programmieralgorithmen
  und Überspringen von leeren Bytes
- Maschinensprache Monitor mit 31 Funktionen
- Modulgenerator für BASIC & Maschinensprache
  Programme mit Autostart
- Eingebautes Betriebssystem mit PP64 Befehlen
  Lesen, Schreiben, LOAD, SAVE, OPEN, CLOSE, PRINT #,.....

Inbetriebnahme

Falls Sie einen SX-64 Executive Computer haben, lesen Sie bitte das Kapitel „PP64 am SX-64“!

Schalten Sie als erstes den Computer aus. Stecken Sie den PP64 EPROM-Programmer in den USER-Port links hinten am Computer. Stecken Sie das PP64 MENÜ-Modul rechts hinten – mit der Schrift nach oben – in den Expansion-Port.

Schalten Sie jetzt den Computer ein. In der linken oberen Ecke des Bildschirms erscheint das EINSCHALTMENÜ:

(1) PP64 MENUE
(2) PP64 ERWEITERUNG
(3) NORMALSYSTEM

Geben Sie jetzt die jeweilige Nummer ein (ohne RETURN-Taste).

Mit (1) kommt Sie in das PP64 MENÜ-Programm; wählen Sie diesen Punkt, um EPROMs zu programmieren etc.

Mit (2) bekommen Sie ein modifiziertes Commodore-Betriebssystem, mit Befehlen zur Handhabung des PP64 als ROM-Disk.

Mit (3) bekommen Sie das normale Commodore-Betriebssystem.

Auf der PP64 Platine befinden sich drei LEDs, welche folgende Bedeutung haben:

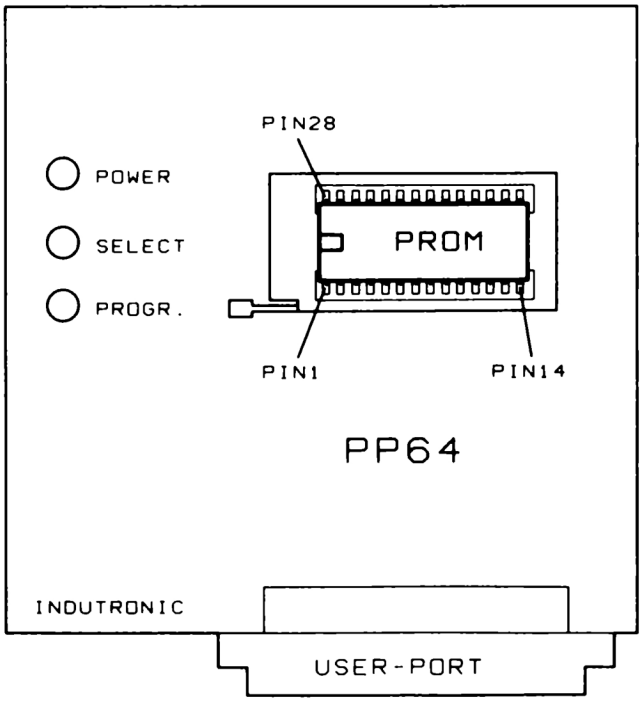
	grün	PP64 ist eingeschaltet
	rot	EPROM steht unter Spannung
	gelb	EPROM wird programmiert
blinkend	gelb	Fehler beim Programmieren

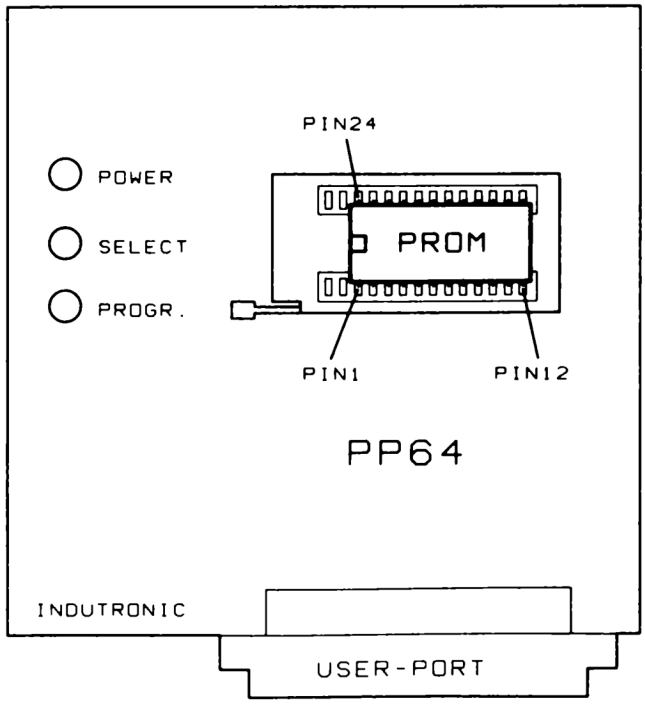
Bei manchen Programmierarten (speziell bei X2864A und X28256A) leuchtet die gelbe LED beim Programmieren nicht!

Beachten Sie bitte, daß beim Einlegen des EPROMs der Hebel nach oben zeigen muß. Stecken Sie dann das EPROM mit der Kerbe nach links in den Sockel. 28 polige EPROMs stecken zur Gänze im Sockel, bei 24 poligen EPROMs bleiben hingegen die ersten 4 Kontakte (links beim Hebel) frei. Legen Sie nun den Hebel nach links um, damit das EPROM fest im Sockel sitzt.

Wenn die rote LED leuchtet, dürfen Sie NIE ein EPROM in den Sockel stecken oder herausnehmen!

Bevor Sie ein EPROM programmieren, informieren Sie sich mit welcher Spannung es programmiert werden soll. Neuere EPROM-Typen werden mit 21 Volt oder soger mit nur 12.5 Volt programmiert. Eine zu große Programmierspannung kann das EPROM zerstören! (Dies gilt vor allem für die Typen 2764, 27128, 27256 und alle C-MOS Typen). Falls Sie sich über die Programmierspannung nicht sicher sind, probieren Sie es zuerst mit 12.5 Volt; falls sofort eine Fehlermeldung kommt, nehmen sie die nächsthöhere Spannung.





PP64 MENÜ PROGRAMM

Nach Wahl von (1) ertönt ein Gong, und das Auswahlmenü für die EPROM-Typen erscheint auf dem Bildschirm (siehe Bild).

PP 64 MENU V1.0 (C) 1985 MERLIN DATA

2516	2532	2564	2716
27C16	2732	2732A	27C32
27C32	2758	2764	2764A
27C64	27C64	27128	27128A
27C128	27C128	27256	27256
27C256	27C256	27512	2815
2816	X2804A	X2816A	X2864A
X28C64A	X28256A	X28C256	48016
5133	5143	5213	52B13
58064	68764	68766	

PROMTYPE 2764 mit 21 VOLT

Mit den Cursor-Tasten wählen Sie Ihren EPROM-/EEPROM-Typ aus. In der untersten Zeile wird die Programmierspannung für das jeweilige EPROM angezeigt. Voreingestellt ist das EPROM vom Typ 2764 mit 21 Volt Programmierspannung. Wenn Sie Ihre Wahl getroffen haben, drücken Sie die RETURN-Taste.
Der Gong bestätigt Ihre Wahl, und es erscheint das Arbeitsmenü (siehe Bild).

!!! Besonderheiten beim 27512 !!!

Dies ist der neueste EPROM-Typ von INTEL mit einer Kapazität von 64 KByte. Dieser EPROM kann von PP64 nur in zwei Durchgängen programmiert werden.

PP64 OHNE SCHALTER:
Um die ersten 32 KByte zu programmieren, müssen Sie eine Verbindung zwischen PIN 1 und PIN 14 herstellen (siehe Bild). Für die zweiten 32 KByte müßen Sie diese wieder entfernen (auch für alle anderen EPROMs).

PP64 MIT SCHALTER:
Links, neben dem Userport-Stecker, befindet sich ein Schiebeschalter der normalerweise in der linken Stellung steht. Für die Programmierung der ersten 32 KByte müssen Sie diesen Schalter nach rechts schieben. Für die zweiten 32 KByte schieben Sie den Schalter nach links.

Oben steht Ihr gewählter EPROM-Typ mit der dazugehörigen Programmierspannung. Darunter die Kapazität in Bytes, die Anfangsadresse im RAM (BRENNSTART), die Endadresse im RAM (BRENNENDE), die Adresse des ersten zu programmierenden Bytes im Prom sowie der gewählte (bzw. vorgeschlagene) Programmieralgorithmus (in REVERS). Die Angaben sind die Hexadezimal.

In der nächsten Zeile erfolgen die Fehler- bzw. Erfolgsmeldungen des Programms in der Form:

STATUS: Meldung

Die nächsten 7 Zeilen enthalten die Kurzbeschreibung der möglichen Funktionen. Um eine Funktion auszuführen, drücken Sie einfach den vor der Beschreibung stehenden Buchstaben (z.B.: R um den Inhalt eines EPROMs in den Speicher zu lesen).
Der gewählte Buchstabe wird in der untersten Zeile angezeigt. Falls Sie diese Funktion ein zweites Mal durchführen wollen, können Sie auch einfach die RETURN-Taste drücken.

Nach Beenden der gewählten Funktion ertönt wieder ein Gong und eine entsprechende Statusmeldung erscheint auf dem Bildschirm.

PP 64 MENU V1.0 (C) 1985 MERLIN DATA

PROMTYP: 2764 mit 21 Volt

KAPAZITÄT : 2000
BRENNSTART : 2000
BRENNENDE : 3FFF
PROMSTART : 0000
ALGORYTHMUS : S/1/2/3

STATUS:

R	PROM LESEN	P	PROM PROGRAMMIEREN
L	VON DISK LADEN	S	AUF DISK SPEICHERN
A	ADRESSE ÄNDERN	T	PROMTYP ÄNDERN
D	DISKVERZEICHNIS	K	DISKKOMMANDOS
B	BLANKCHECK	V	VERGLEICHEN
G	MODULGENERATOR	M	MONITOR
X	BEENDEN		

IHRE WAHL : B

R PROM LESEN

Mit dieser Funktion wird der Inhalt des PROMs in den Speicher gelesen. Falls kein Prom eingelegt ist, wird Hex FF eingelesen. Vor dem Einlesen können Sie mit der Funktion „A Adressen ändern“ festlegen wieviele Bytes und ab welcher Adresse eingelesen werden sollen.
Mit dieser Funktion können Sie auch einige ROMs und PROMs einlesen, die nicht mit dem PP64 programmierbar sind.

P PROM PROGRAMMIEREN

Hiermit wird der Inhalt des Speichers in das EPROM programmiert. Vor der Programmierung eines Bytes wird geprüft, ob dieses gleich mit dem jetzigen

EPROM-Inhalt ist. Bei Gleichheit geht das Programm sofort zum nächsten Byte. Dadurch wird die Programmierzeit weiter verringert. Vor dem Programmieren können Sie mit der Funktion „A Adresse ändern" festlegen, wieviele Bytes und ab welcher Adresse sie programmiert werden sollen.

Tritt beim Programmieren ein Fehler auf, so wird der Vorgang abgebrochen und die entsprechende Speicheradresse angezeigt.

L VON DISK Laden

Nach der Eingabe von „L" wird der Bildschirm gelöscht und das Programm fragt Sie nach dem Namen des zu ladenden Diskfiles:

NAME :

Geben Sie hier den gewünschten Namen ein und drücken Sie die RETURN-Taste (falls Sie abbrechen wollen, geben Sie „←" als Namen ein). Wenn keine Datei mit diesem Namen existiert oder sonst ein Fehler auftritt, wird dies in der darunterliegenden Zeile angezeigt. Nach Drücken der RETURN-Taste erscheint wieder das Arbeitsmenü. Ansonsten erscheint die Meldung: LADE XXXX wobei anstelle der XXXX die momentane Ladeadresse angezeigt wird (geladen wird immer ab der BRENNSTART Adresse). Falls die Datei länger als der eingestellte RAM-Bereich ist, wird bei Erreichen der BRENNENDE Adresse abgebrochen (für größere Programme den PROM-Typ ändern). Nach erfolgreichem Laden der Daten erscheint wieder das Arbeitsmenü und die Endadresse steht bei BRENNENDE.

S AUF DISK SPEICHERN

Nach der Eingabe von „S" wird der Bildschirm gelöscht und das Programm fragt Sie nach dem Namen, unter dem es auf Disk gespeichert werden soll (falls Sie abbrechen wollen, geben Sie „←" als Namen ein). Geben Sie den Namen ein und drücken Sie die RETURN-Taste. Wenn der Name schon existiert oder keine Diskette eingelegt ist, erfolgt eine entsprechende Fehlermeldung. Andernfalls kommt die Meldung: SCHREIBEN. Danach erscheint wieder das Arbeitsmenü.

Gespeichert wird immer der Speicher von BRENNSTART bis BRENNENDE!

A ADRESSEN ÄNDERN

Hier können Sie die für die Funktionen R, P, L, S, B und V benötigten Adressen ändern. Beachten Sie bitte, daß alle Eingaben in Hexadezimal sind und immer alle 4 Stellen eingegeben werden müssen. Falls die angezeigte Adresse nicht geändert werden soll, drücken Sie einfach die RETURN-Taste.

BRENNSTART	ist die Anfangsadresse des Speichers. Der eingegebene Wert muß zwischen HEX 2000 und 9FFF liegen.
BRENNENDE	ist die Endadresse des Speichers. Der eingegebene Wert kann nicht kleiner als BRENNSTART und nicht größer als BRENNSTART+KAPAZITÄT-1 sein (maximal HEX 9FFF).

PROMSTART ist die Adresse des ersten Bytes im EPROM ab der programmiert oder gelesen werden soll. Der eingegebene Wert kann nicht größer als KAPAZITÄT+BRENNSTART-BRENNENDE-1 sein.

Diese Bereiche werden bei der Eingabe überprüft und bei Überschreiten zurückgewiesen. Anschließend müssen Sie noch den Programmieralgorithmus eingeben. Drücken Sie dafür die Tasten S, 1, 2 oder 3.

S Standard Programmierungsart
1 Intelligent #1 (Jason-Ranheim Co.)
2 Intelligent #2 (Intel Nr. 1)
3 Intelligent #3 (Intel Nr. 2)

Normalerweise sollten Sie aber die empfohlene Programmierart (in REVERS angezeigt) benutzen.

T PROMTYP ÄNDERN

Damit gelangen Sie wieder in das Auswahlmenü. Wählen Sie mit den Cursor-Tasten den gewünschten Typ und bestätigen Sie mit der RETURN-Taste.

D DISKVERZEICHNIS

Auf dem Bildschirm erscheint das Inhaltsverzeichnis der Diskette im Laufwerk Nr. 8.

K DISK-KOMMANDOS

Der Bildschirm wird gelöscht, und das Programm fragt Sie nach dem Diskkommando:

BEFEHL:

Geben Sie hier den gewünschten Befehl ein. Falls die Diskettenstation einen Fehler meldet, wird dieser angezeigt; ansonsten kommt wieder das Arbeitsmenü. Folgende Befehle sind möglich:

I	Initialisieren	I
S	File löschen	S:NAME
R	Namen ändern	R:NEUNAME=ALTNAME
V	Disk aufräumen	V
C	File kopieren	C:NEUFILE=ALTFILE
N	Disk löschen	N:DISKNAME
	Disk formatieren	N: DISKNAME,ID

Beim „S"-Befehl können Sie natürlich auch einen JOKER (*) angeben.

Weiteres dazu entnehmen Sie bitte ihrem Diskettenhandbuch.

B BLANKCHECK

Dieser Befehl überprüft ob das EPROM leer ist (im Bereich BRENNSTART-BRENNENDE). „Leer" bedeutet, daß alle Bytes den Wert HexFF haben. Falls

das EPROM leer ist, erfolgt die Meldung: KEINE FEHLER. Ansonsten: NICHT LEER: XXXX
Wobei XXXX die Anzahl der fehlerhaften Bytes ist.

V VERGLEICHEN

Mit diesem Befehl können Sie den Speicher mit dem Inhalt eines EPROMs vergleichen. Beachten Sie dabei die mit „Adressen ändern" eingestellten Werte. Falls alle Bytes gleich sind, erfolgt die Meldung: KEINE FEHLER. Ansonsten: DATEN VERSCHIEDEN: XXXX
Wobei XXXX die Anzahl der nicht identischen Bytes ist.

G MODULGENERATOR

Nachdem der Bildschirm gelöscht wurde, fragt Sie das Programm nach dem Namen. Geben Sie den Namen des Programms ein. Anschließend wird das Programm geladen. Falls das Programm größer als die EPROM-Kapazität ist (Programmgröße plus Autostart Routine), erfolgt eine Fehlermeldung; ansonsten erscheint das Arbeitsmenü, und Sie können mit „P" das Programm direkt programmieren.

Falls das eingelesene Programm keine BASIC-Startadresse hat (ungleich Hex 0801 bzw. Dezimal 2049), wird dies gemeldet:

KEIN BASICPROGRAMM, SYS – STARTADRESSE EINGEBEN:

Geben Sie hier mit Hex-Ziffern die Startadresse (4 Stellen) ein. Falls Sie die Adresse nur in Dezimal kennen, können Sie im Maschinensprache-Monitor diese in eine Hexadezimalzahl umwandeln. Der Befehl dazu lautet: #zahl
(Drücken Sie M, geben Sie dann das Doppelkreuz gefolgt von der Dezimalzahl ein und drücken Sie RETURN. Notieren Sie sich das Ergebnis und drücken Sie X RETURN; damit gelangen Sie wieder in das Auswahlmenü).

Falls Sie die benötigte EPROM-Größe berechnen wollen, laden Sie mit LOAD Ihr Programm ganz normal und geben Sie anschließend folgende Zeile ein:

PRINT PEEK (45)+256*PEEK (46)-2048+122

Das Ergebnis dieser Rechnung ist die benötigte EPROM-Größe. Bis zu einem Wert von 16384 können mit dieser Funktion Autostart-Module erstellt werden.

M MONITOR

Mit diesem Befehl gelangen Sie in den eingebauten Maschinensprache Monitor.

X BEENDEN

Mit diesem Befehl verlassen Sie das PP64 MENÜ und kommen wieder zum Einschaltmenü.

PP64 ERWEITERUNG

Nach Wahl von (2) erscheint die Einschaltmeldung des PP64-Expanders. Dieser ersetzt das Commodore-Betriebssystem. Bis auf folgende Änderungen bleibt die Funktion des Commodore-Betriebssystems voll erhalten:

1) (SHIFT) + (RUN/STOP) lädt das Inhaltsverzeichnis von Laufwerk Nr. 8 und listet es am Bildschirm.
2) (CTRL) + (COMMODORE) + (SHIFT) macht einen RESET.
3) Die Standard-Gerätenummer ist 8. Damit brauchen Sie bei LOAD, SAVE und VERIFY keine Nummer angeben (LOAD „NAME" (RETURN)).
4) Der LOAD-Befehl kann direkt im aufgelisteten Inhaltsverzeichnis gegeben werden. Nachfolgender Text wird ignoriert.
5) Die Routinen für den Kassettenrekorder und die RS-232 Schnittstelle sind nicht vorhanden.

NEUE BEFEHLE IM DIREKTMODUS

Z setzt die PP64 Software in den Grundzustand (nach Fehlern etc eingeben).

£ MEMSTART, MEMENDE, ROMSTART, KW
Daten aus dem EPROM ab Adresse ROMSTART in den Computer in den Bereich MEMSTART bis MEMENDE einlesen. KW ist das Kontrollwort für das jeweilige EPROM.
Beispiel: Einlesen eines 2764 EPROMs von HEX 4000: £ 16384, 24575, 0, 5

π MEMSTART, MEMENDE, ROMSTART, KW, PW
Programmiert die Daten im Bereich MEMSTART bis MEMENDE ins EPROM ab Adresse ROMSTART. KW ist das Kontrollwort für das EPROM und PW gibt die Programmierart an.
Beispiel: Programmieren eines 2764 EPROMs von HEX 4000:
π 16384, 24575, 0, 5, 10

(SHIFT)+(L)NAME:KW
Liest ein Maschinensprache-Programm aus dem EPROM in den Speicher von Adresse MEMSTART bis MEMENDE ein.

(SHIFT)+(S)NAME:KW, MEMSTART, MEMENDE, PW
Schreibt ein Maschinensprache Programm von Adresse MEMSTART bis MEMENDE ins EPROM.

$KW Auflisten des EPROM-Inhaltsverzeichnisses (jede Zeile muß durch einen Tastendruck abgerufen werden). Die Anzeige erfolgt mit folgendem Format:

xxxxx Programmname tt

wobei xxxxx die Startadresse im EPROM angibt und tt den Dateityp. P steht für Programme, D für Dateien. Ein nachgestellter * (zB: P*) bedeutet, daß das File nicht geschlossen wurde.

⟨SHIFT⟩+⟨E⟩ dient zum Löschen von EEPROMs (WICHTIG: DARF NICHT FÜR X2864A, X28256A, X28C64 UND X28C256 VERWENDET WERDEN !!!).

NEUE BASIC BEFEHLE

LOAD"NAME:KW",16 Einlesen eines BASIC Programmes

SAVE"NAME:KW",16,PW Schreiben eines BASIC Programmes

OPEN fn,16,PW, "NAME:KW" Eröffnen einer Datei

GET#fn,.... nächstes Byte lesen

PRINT#fn,... Schreiben

INPUT#fn,... Lesen

CMD#fn Ausgabe vom Bildschirm zum EPROM umlenken

CLOSE fn Schließen der Datei

ANMERKUNGEN

Namen, die in spitzen Klammern(⟨⟩) stehen, bezeichnen Tasten des C=64

MEMSTART Anfang des Speicherbereichs im Computer

MEMENDE Ende des Speicherbereichs

ROMSTART Startadresse im EPROM
(MEMENDE-MEMSTART+1 ergibt die Anzahl Bytes)

KW Kontrollwort, bestimmt den EPROM-Typ (siehe Tabelle).

PW Programmierwort, bestimmt die Programmierart (siehe Tabelle.)

NAME Filename, maximal 16 Zeichen
Beim LOAD-Befehl kann der Name auch abgekürzt werden:
LOAD"Test*:5",16
(Hier wird das erste Programm, bei welchem die ersten 4 Buchstaben TEST lauten, geladen.
Mit LOAD"*:5",16 wird das erste Programm aus dem EPROM geladen.)

Alle Zahlenwerte müssen in DEZIMAL eingegeben werden!

TABELLE FÜR KONTROLLWORT (KW) UND PROGRAMMIERWORT (PW)

			PROGRAMMIERWORT			
TYPE	Vpp	KW	STD	I#1	I#2	I#3
2316		40	NUR LESEN!			
2332		24	NUR LESEN!			
2364		48	NUR LESEN!			
2516	25	40	3	7	10	14
2532	25	24	3	7	10	14
2564	25	20	3	7	10	14

			PROGRAMMIERWORT			
TYPE	Vpp	KW	STD	I#1	I#2	I#3
2758	25	40	3	7	10	14
2716	25	40	3	7	10	14
27C16	25	40	3	7	10	14
2732	25	224	3	7	10	14
2732A	21	225	3	7	10	14
27C32	25	224	3	7	10	14
27C32	21	225	3	7	10	14
2764	21	5	3	7	10	14
2764A	12.5	6		7	10	14
27C64	21	5		7	10	14
27C64	12.5	6		7	10	14
27128	21	5	3	7	10	14
27128A	12.5	6		7	10	14
27C128	21	5		7	10	14
27C128	12.5	6		7	10	14
27256	21	229		6		14
27256	12.5	230		6		14
27C256	21	229		6		14
27C256	12.5	230		6		14
27512	12.5	226		6		14
68764	25	48		6	11	15
68766	25	48		6	11	15
48016	25	40	2	5	9	13
5133	21	5	3	7		15
5143	21		3	7		15
			EEPROM			
X2804A	21	57	1			
X2816A	21	57	1			
X2864A	.5	23	1			
X28C64	5	23	1			
X28256A	5	23	1			
X28C256	5	23	1			
2815	21	57	3	6	11	15
2816	21	57	3	7		
58064	5	7		7		
5213	21	57	2	4	9	13
52B13	21	57	1			

SPEICHERFORMAT VON DATEIEN IM EPROM

Ein PP64 File im EPROM besteht aus einem 21 Byte großen Header gefolgt von den eigentlichen Daten. Der Header hat folgendes Format:

	File Status	Adresse letztes Byte		Start Adresse		File Name	Daten
Byte	0	1	2	3	4	5-20	21–

File Status: dieses Byte gibt Auskunft über die Gültigkeit des Eintrags und den Status des Files.

- Bit 7 wird durch OPEN auf 0 gesetzt
- Bit 6 wird durch CLOSE auf 0 gesetzt
- Bit 5,4,3 werden nicht verwendet
- Bit 2 wenn 0 ist das File schreibgeschützt
- Bit 1 wenn 0 ist das File ABSOLUT zu laden
- Bit 0 wenn 0 ist das File ein Programm

Adresse letztes Byte: zeigt auf das letzte belegte Byte (1=low, 2=higher Byte der Adresse), nächstes Byte enthält den File Status der nächsten Datei oder HEX FF.

Start Adresse: Falls im File Status die Bits 0 und 1 den Wert 0 haben, so wird das Programm auf diese Adresse geladen.

File Name: 16 Bytes, welche den Filenamen enthalten. Unbenutzte Bytes haben den Wert HEX FF!

Der PP64 sucht nach jedem SAVE-Befehl sofort alle Header nach einem gleichen Namen ab. Findet er den gleichen Namen, wird der Befehl nicht ausgeführt.

Bei einem OPEN-Befehl sucht der PP64, ob der Name schon existiert und eröffnet es falls vorhanden. Ansonsten wird ein neuer Header angelegt.

Der PP64 unterstützt einen Daten-Zeiger, welcher immer auf das nächste zu schreibende Byte zeigt. Maximal kann dieser Wert 32767 haben.

PROGRAMMIERUNGSARTEN

S- Standardprogrammierung:

Hierbei wird jedes Byte des EPROMs mit einem Programmierimpuls konstanter Dauer programmiert. Es wird hierbei die gleiche Pulsdauer, die der Hersteller bei den Programmiertests verwendet, programmiert. Bei diesem Verfahren wird der Fall des längsten Programmiervorganges bei jedem Byte angenommen. Da nicht alle Bytes die gleiche Zeit zur Programmierung benötigen, ist dieses Verfahren zeitlich nicht optimal. Programmiert man beispielsweise einen 2764 nach der Standardmethode (Pulsdauer 48 ms PW=3), benötigt man für die 8192 Bytes ca 7 min.

Programmierwort	Pulsdauer (in Millisekunden)
0	6
1	12
2	24
3	48

Bei den folgenden Verfahren wird der Umstand genutzt, daß nicht alle Bytes die gleiche Programmierzeit benötigen. Es wird zunächst mit einem kurzen Impuls programmiert und überprüft, ob das Byte übernommen wurde. Trifft dies zu, wird das Byte nochmals mit einem „ÜBERPROGRAMMIER-IMPULS", dessen Länge von der Anzahl und Länge der Testimpulse abhängt, nachprogrammiert. Wird auch nach einer Anzahl von Testimpulsen bzw. nach dem überprogrammier-Impuls die Information nicht übernommen, so wird der Programmiervorgang abgebrochen.

1- Intelligentes Verfahren Nr. 1

Bei dieser Programmierart wird mit einer Impulsdauer von 0.1 ms begonnen. Nach jedem Versuch wird die Impulsdauer verdoppelt, bis zu einem Wert von 100 ms. Anschließend wird mit der Summe der Impulse nachprogrammiert. Die maximale Impulsdauer ist dann ca. 240 ms.

2- Intelligentes Verfahren Nr. 2

Dies ist ein von INTEL empfohlenes Verfahren zur Programmierung der Typen 2732A, 2764 und 27128.
Hier wird ebenfalls nach jedem Testimpuls der Programmiererfolg kontrolliert. Ist der Vergleich negativ, wird der nächste Testimpuls mit gleicher Dauer angelegt. Bei positivem Ergebnis wird mit der vierfachen Zeit nachprogrammiert. Ist auch nach 10 Testimpulsen kein positives Ergebnis da, so wird mit einem 60 ms-Impuls nachprogrammiert. Ergibt sich auch jetzt keine Übereinstimmung, wird der Programmiervorgang abgebrochen.

3- Intelligentes Verfahren Nr. 3

Dies ist ein von INTEL empfohlenes Verfahren zur Programmierung der Typen 27128A, 27256 und 27512.
Dieses Verfahren ist ähnlich dem obengenannten. Der Nachprogrammierimpuls hat aber nur die dreifache Zeit. Ist nach maximal 15 Testimpulsen der Vergleich negativ, wird der Programmiervorgang abgebrochen.

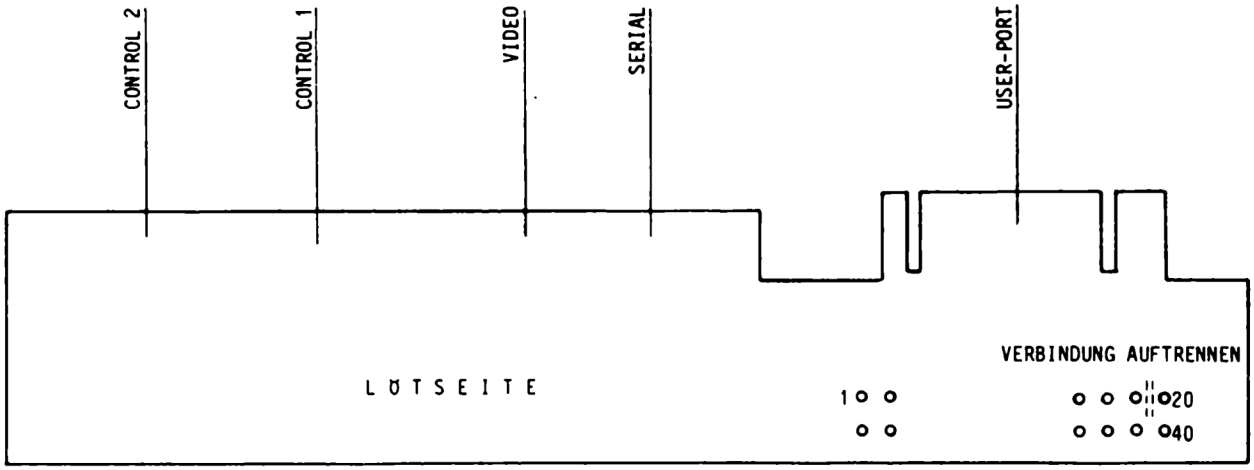
PP64 UND SX-64

Der PP64 erzeugt die für die Programmierung notwendige Spannung von 25 (bzw. 21 oder 12.5) Volt durch Verdreifachung der vom C=64 potentialfrei gelieferten 9 Volt Wechselspannung. Beim SX-64 ist diese Spannung auch am USERPORT vorhanden, jedoch NICHT potentialfrei ausgeführt, sondern einseitig mit der Masse des SX verbunden. Nach Auftrennen dieser Verbindung ist der PP64 auch mit dem SX-64 verwendbar.

UMBAUVORSCHLAG

Vor dem im folgenden beschriebenen Eingriff ist unbedingt der Netzstecker zu ziehen, und alle Peripheriegeräte sind abzustecken.

Entfernen Sie die obere Abdeckung des SX. Dadurch wird die Platine 'CONNECTOR PCB', auf der sich die Stecker für USERPORT, Serialport etc befinden, zugänglich. Drehen Sie die drei Befestigungsschrauben heraus und heben Sie die Platine hoch. Durchtrennen Sie auf der Lötseite gemäß Skizze die Verbindung zwischen Pin 19 und 20. Damit ist die 9 Volt Versorgung potentialfrei gemacht.



MASCHINENSPRACHE-MONITOR

	Inhalt		Seite
A	Assemble	Assemblieren	18
B	Breakpoint	Programmstop bei Simulator	19
C	Compare Memory	Speicher-Vergleich	17
D	Disassemble	Programm zeigen	17
F	Fill Memory	Speicher mit Wert füllen	16
G	Go	Programm starten	18
H	Hunt Memory	Speicher durchsuchen	17
I	Ascii Display	Speicher-Anzeige ASCII	16
L	Load	Programm laden	15
M	Memory Display	Speicher-Anzeige HEX	16
N	New Locator	Absolutadressen ändern	20
Q	Quick Trace	Programm-Simulator	19
R	Register	CPU Register anzeigen	15
S	Save	Programm speichern	15
T	Transfer Memory	Speicher kopieren	16
V	Verify	Speicher mit Disk vergleichen	16
W	Walk	Einzelschritt	18
X	eXit	Rücksprung zum PP64 MENUE	19
EC	Edit Character	Zeichensatz als BITMUSTER	19
ES	Edit Sprite	Sprite als BITMUSTER	20
ZB	Copy RAM A000-BFFF	RAM unter BASIC kopieren	17
ZK	Copy RAM E000-FFFF	RAM unter KERNAL kopieren	17
ZZ	Copy Zeichensatz	Zeichensatz ins RAM kopieren	16
#	Decimal -> HEX	Umrechnung Dezimal -> HEX	19
$	HEX -> Decimal	Umbrechung HEX -> Dezimal	19
,	modify Program in HEX	Programm in HEX ändern	17
;	modify Register	Register ändern	15
:	modify Memory	Speicher-Daten ändern	16
(	modify Character	Zeichensatz ändern	20
)	modify Sprite	Sprite ändern	20

* ,vor einen Befehl gesetzt schaltet auf Drucker Ausgabe z.B.:

*M A000 BFFF

listet den Inhalt des BASIC ROM auf den Drucker.

Der Maschinensprache Monitor arbeitet **NICHT** mit dem PP64-EXPANDER Betriebssystem, sondern mit dem **NORMALEN** Commodore Betriebssystem. Falls Sie ein anderes Betriebssystem haben, so benutzt der Monitor natürlich dieses auch (z.B.: Speeddos oder Hypraload).

Nach dem Aufruf des Monitors wird der Bildschirm gelöscht und der Monitor zeigt die aktuelle Registerinhalte an:

```
B*
      PC   NMI  AC   XR   YR   SP   NV#BDIZC
.;    C04C FE47 00   C3   00   F6   ..**..*.
```

Die gleiche Anzeige bekommen Sie auch mit dem **R- Befehl.** Gezeigt wird von links nach rechts:

PC	Programmzähler
NMI	Adresse der NMI Routine (Restore)
AC	Akkumulator Inhalt
XR	X-Register
YR	Y-Register
SP	Stapelzeiger

Das Prozessorstatus-Register wird binär angezeigt, wobei *eine 1 bedeutet und . eine 0.

Zum Ändern der Registerinhalte können Sie direkt mit dem Cursor in die Zeile gehen und die alten Werte überschreiben. Vergessen Sie aber nicht, anschließend die RETURN-Taste zu drücken.

Beachten Sie bitte, daß der Wert unter NMI **NICHT** änderbar ist. Wenn Sie diesen modifizieren wollen, müssen Sie das direkt auf Adresse $0318/$0319 machen.

L Programm Laden

.L "NAME"

lädt das Programm NAME von der Diskstation Nr. 8 auf die gleiche Adresse wie beim Speichern (BASIC Programme z.B.: HEX 0801).

.L "NAME",09

lädt das Programm vom Gerät mit der Nummer 9.

.L "NAME",08,2000

lädt das Programm ab der Adresse HEX 2000 in den Speicher.

S Programm Speichern

.S „NAME",08,2000,4000

speichert den Bereich 2000 bis 3FFF inklusive auf die Diskettenstation Nr.8. Beachten Sie bitte, daß die Endadresse immer um 1 größer sein muß.

V Programm Verify

.V "NAME"

vergleicht, ob das Programm NAME auf der Diskette gleich dem im Speicher ist.

.V "Name",09

wie oben, aber Gerät Nr. 9

.V "NAME",08,200

vergleicht das Programm NAME mit dem Speicherbereich ab HEX 2000.

M Speicheranzeige

.M 2000 2010

zeigt den Speicherinhalt von HEX 2000 bis 2017. Dabei werden die Inhalte in HEX und als Zeichen angezeigt; jeweils 8 Bytes in einer Zeile.
Die angezeigten Werte können Sie direkt am Bildschirm ändern, damit sie gespeichert werden müssen Sie die RETURN-Taste drücken.

Eingeben können Sie nur HEX-Werte, keine Zeichen! (Diese werden nur angezeigt!).

I Speicheranzeige

siehe M Befehl

: Daten Eingeben

.: 2000 01 02 03 04 05 06 07 08

nach der Eingabe von Doppelpunkt und Adresse können Sie bis zu 8 Datenbytes eingeben, welche ab der Adresse in den Speicher geschrieben werden.
Falls an der angegebenen Adresse kein Speicher existiert, wird das durch ein ? angezeigt.

F Speicher füllen

.F 2000 2FFF 2A

füllt den Speicher von HEX 2000 bis 2FFF inklusive mit dem Wert 2A auf. Wenn der Wert nicht auf die angegebene Adresse geschrieben werden kann, wird diese Adresse angezeigt und die Ausführung abgebrochen.

T Speicher kopieren

.T 2000 3FFF 6000

kopiert den Speicher von HEX 2000 bis 3FFF inklusive auf den Speicherbereich ab 6000. Bei einem Schreibfehler wird die jeweilige Adresse am Bildschirm ausgegeben, die Ausführung aber nicht unterbrochen.

ZZ Zeichensatz kopieren

.ZZ 2000

kopiert den C=64 Zeichengenerator auf die Adresse HEX 2000 bis 2FFF (da der Zeichengenerator 4 kByte groß ist).

Den Zeichensatz kann man dann mit den Befehlen 'EC' und '(' anschauen bzw. modifizieren.

ZB RAM unter BASIC kopieren

.ZB 2000

kopiert das unter dem BASIC ROM liegende RAM (8 kByte) auf die Adresse HEX 2000

ZK RAM unter KERNAL kopieren

.ZK 2000

kopiert das unter dem KERNAL ROM liegende RAM (8 kByte) auf die Adresse HEX 2000

C Speicher vergleichen

.C 2000 3FFF 6000

vergleicht den Speicher von HEX 2000 bis 3FFF inklusive mit dem Bereich ab 6000. Bei ungleichen Inhalten wird die jeweilige Adresse angezeigt.

H Speicher durchsuchen

.H A000 BFFF 'ERROR

durchsucht den Speicher von A000 bis BFFF nach Test: ERROR. Alle Übereinstimmungen werden durch Ausdruck der Anfangsadresse angezeigt.

.H 1000 1200 A9 00

wie oben, aber statt dem Text wird nach einer Bytefolge (hier: A9 00 gesucht.

D Programm anzeigen

.D 1000 1004

listet das Programm von HEX 1000 bis 1004 am Bildschirm:

```
.,   1000   A9   00        LDA   #$00
.,   1002   85   FA        STA   $FA
.,   1004   4C   E2   FC   JMP   $FCE2
```

, Programm ändern

bei obigem Programmbeispiel kann man direkt auf dem Bildschirm die in HEX angezeigten Befehle ändern. Die Memonische Form (wie LDA etc.) kann nicht geändert werden. Sie dient nur der besseren Lesbarkeit.
Programme, die in Hex vorliegen kann man auch direkt eingeben:

```
., 2000  A9  2A
., 2002  20  D2  FF
., 2005  00
,. 2006
```

A Programm eingeben

```
.A 2000 LDA #$2A
.A 2002 JSR $FFD2
.A 2005 BRK
.A 2006
```

Bei der Programmeingabe müssen Sie die Daten oder Adressen immer in Hexadezimal mit vorgestelltem $-Zeichen angeben. Die Adressierungsarten werden wie folgt angegeben

Adressierung:	Format:	Beispiel:
Akkumulator	nichts	ASL
Immediate	#$aa	LDA #$2A
Zero Page	$aa	STA $FA
Absolut	$aaaa	JSR $FFD2
ZP Indexed	$aa,Y	LDA $00,Y
	$aa,X	LDA $00,X
Absolut Indexed	$aaaa,Y	LDA $1000,Y
	$aaaa,X	LDA $1000,X
Indirekt	($aaaa)	JMP ($0300)
Indexed Indirekt	($aa,X)	LDA ($00,X)
Indirekt Indexed	($aa),Y	STA ($FA),Y
Relative	$aaaa	BEQ $1000

Falls Sie ein mit dem 'D'-Befehl erstelltes Programmlisting am Bildschirm ändern wollen, können Sie dies auch mit dem 'A'-Befehl tun: ändern Sie einfach den ',' in der 2. Spalte in ein 'A' und geben Sie den neuen Befehl ein.
Wenn Sie nach dem neu eingegebenen Befehl einen Doppelpunkt eingeben, wird ein eventuell folgender Text ignoriert.

G Programm ausführen

```
.G
```

Das Programm, welches an der im 'PC' stehenden Adresse liegt, wird ausgeführt. Am Programmende sollte ein BRK Befehl stehen (kein RTS).

```
.G FCE2
```

Das Programm ab Adresse Hex FCE2 wird gestartet.

W Einzelschritt (Trace)

```
.W 2000
```

Der Befehl auf der Adresse Hex 2000 wird ausgeführt und anschließend werden die Register und der jeweilige Befehl angezeigt:

```
AC XR YR SP NV#BDIZC PC Befehl
```

nach Drücken der Leertaste wird der nächste Befehl ausgeführt. Die Funktion können Sie mit der STOP-Taste abbrechen.

Q Programmsimulator (Quicktrace)

.Q 2000

Das Programm ab Hex 2000 wird Befehl für Befehl ausgeführt. Bei betätigen der STOP-Taste wird die Ausführung beendet. Eine zweite Möglichkeit, das Programm zu beenden, ist die Breakpoint-Funktion.

B Breakpoint setzen

.B 2700 0007

Ein mit 'Q' gestartetes Programm wird gestoppt, wenn der Befehl auf Adresse 2700 zum 7. Mal ausgeführt wird.

.B 2700

Ein mit 'Q' gestartetes Programm wird mit Erreichen dieser Adresse gestoppt.

Nachdem das Programm durch einen Breakpoint gestoppt wurde, kann wie mit dem 'W'-Befehl weitergearbeitet werden.

 # Dezimal nach HEX

.#32767

Nach Eingabe dieses Befehls wird die angegebene Dezimalzahl in eine HEX-Zahl umgerechnet:

.#32767=7FFF

$ HEX nach Dezimal

.$7FFF

Nach dieser Eingabe wird die HEX-Zahl in eine Dezimalzahl umgerechnet:

.$7FFF=32767

Als Option können Sie hier auch HEX-Zahlen addieren oder subtrahieren:

.$0010+1000−0004

ergibt:

.$0010+1000−0004= 4108

X Rücksprung zum PP64 MENUE

.X

EC Speicher binär zeigen

.EC 2000 2007

Nach Eingabe dieses Befehls werden 8 Bytes binär angezeigt:

.(2000 ....****
.(2001 ****....
.(2002 ....****
.(2003 ****....

.(2004 ....****
.(2005 ****....
.(2006 ....****
.(2007 ****....

Ein '*' bedeutet dabei ein gesetztes und ein '.' ein gelöschtes Bit. Auf einfache Weise können Sie hier Ihren eigenen Zeichensatz generieren.

ES Speicher binär zeigen

.ES 2000 203F

Dieser Befehl ist ähnlich dem 'EC'-Befehl, es werden allerdings immer 3 Byte in einer Zeile dargestellt:

.) 2000 ********.....********
.) 2003 .....********.....
usw....

(Byte binär eingeben

siehe 'EC'-Befehl.

) 3 Byte binär eingeben

siehe 'ES'-Befehl.

N Programm-Adressen ändern

dient zum Verschieben eines Programms an einen anderen Speicherplatz. Ein Programm kann aus drei verschiedenen Teilen bestehen: Programmcode, Programmdaten, Adresstabellen.

Zusätzlich gibt es noch drei verschiedene Adressbereiche:
Absolute Adressen kleiner als der Programmstart
Absolute Adressen größer als das Programmende
Adressen innerhalb des Programms
Nur die letzteren müssen geändert werden.

Um ein Programm, das von 1000 bis 1FFF geht und Daten (z.B.: Texte von 1D00 bis 1EFF) sowie eine Adresstabelle (von 1F00 bis 1FFF) hat, auf die Adresse 8000 zu verschieben, müssen wir wie folgt vorgehen:

.T	1000	1FFF	8000				Gesamtes Programm kopieren
.N	8000	8CFF	7000	1000	1FFF		Programmadresse umrechnen
.N	8F00	8FFF	7000	1000	1FFF	W	Adresstabelle umrechnen

Damit ist das Programm jetzt auf der Adresse 8000 lauffähig.

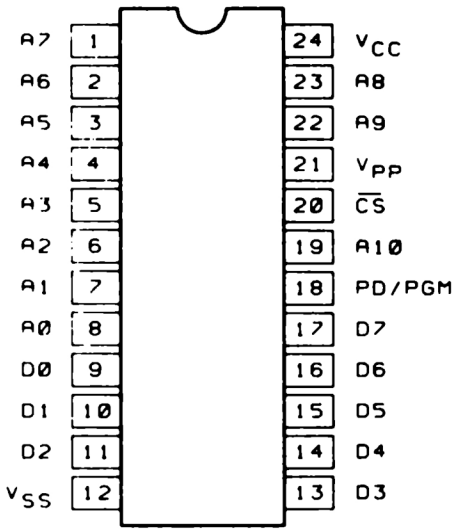
.N 1111 2222 3333 4444 5555

1111 gibt den Programmstart an, 2222 das Programmende. 3333 gibt den Abstand zum ursprünglichen Programm an (Offset). 4444 gibt die unterste zu ändernde Adresse an (damit keine Zero Page Adressen geändert werden) und 5555 die oberste zu ändernde Adresse.

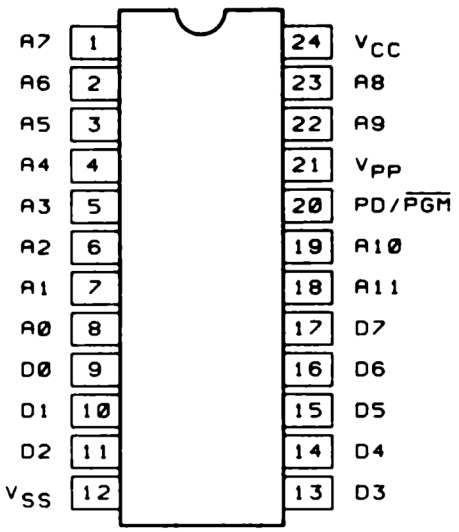
Wenn dem Befehl ein W nachgestellt wird, werden 16-Bit Adressen umgerechnet (Lower – Higher Byte).

PINBELEGUNG DER VERWENDBAREN PROMS 1

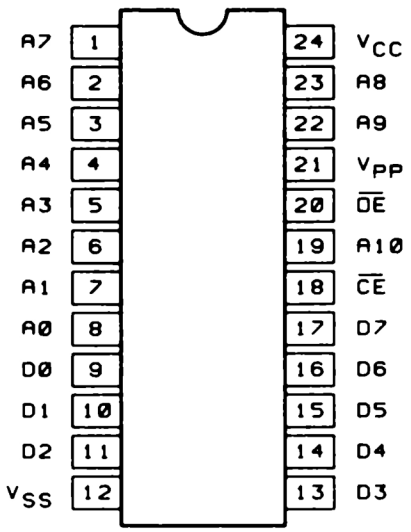
2516



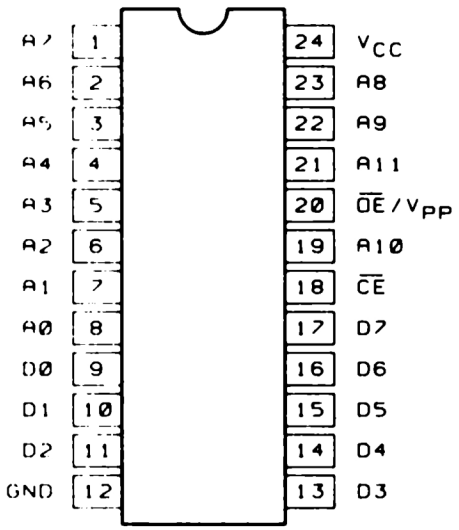
2532



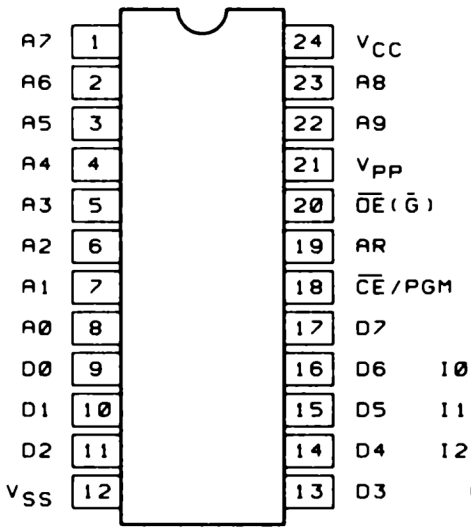
2716



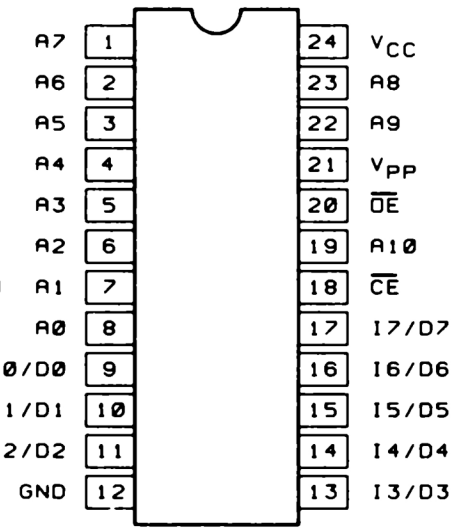
2732



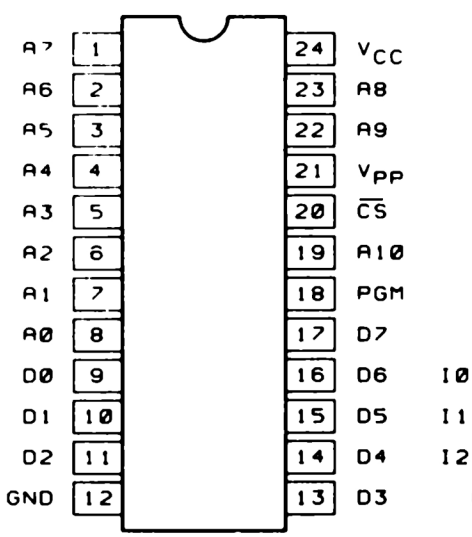
2758



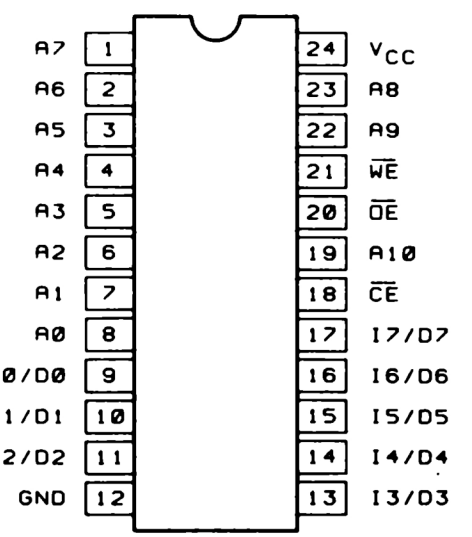
2815/2816



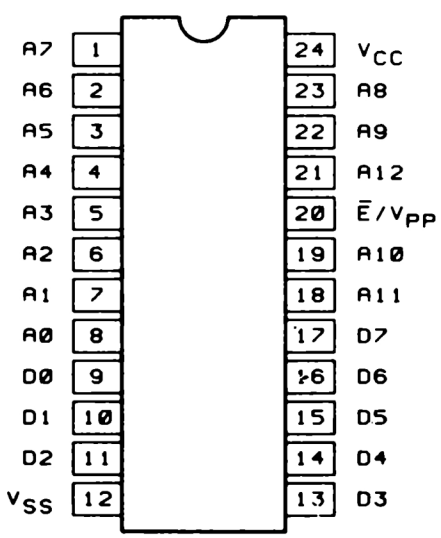
48016



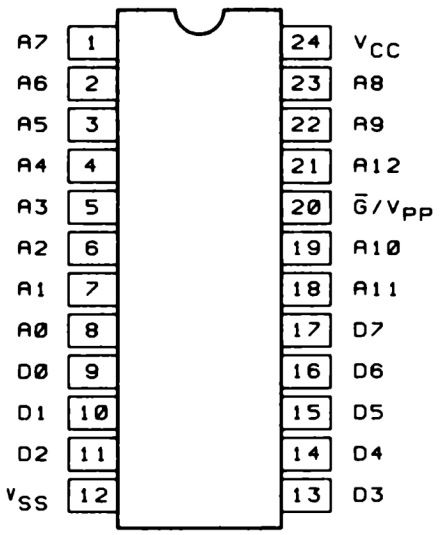
5213/52813



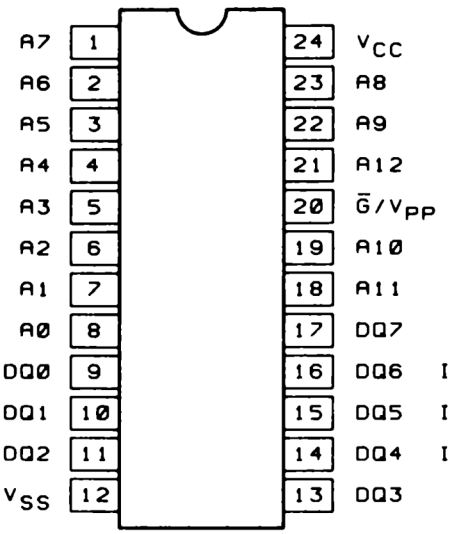
68764



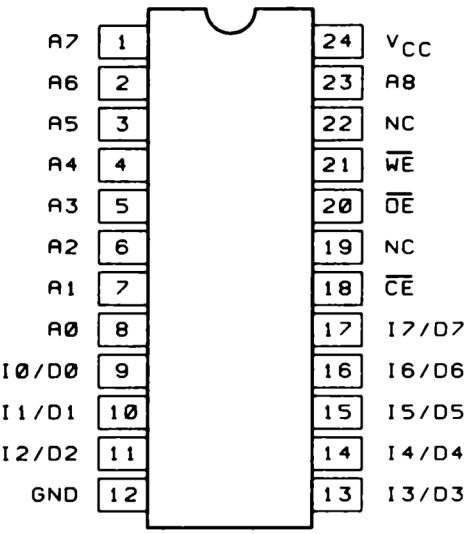
68766



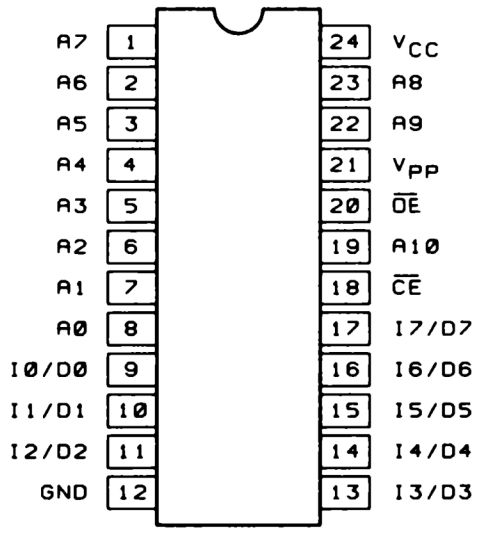
68769



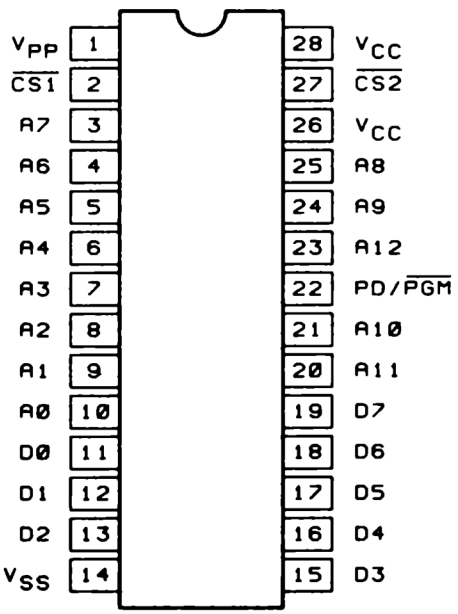
X2804A



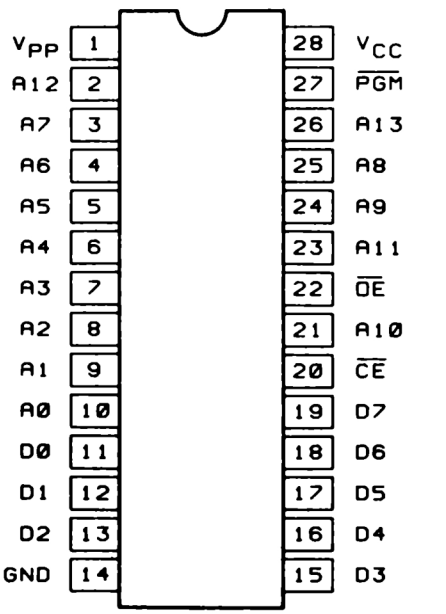
X2816A



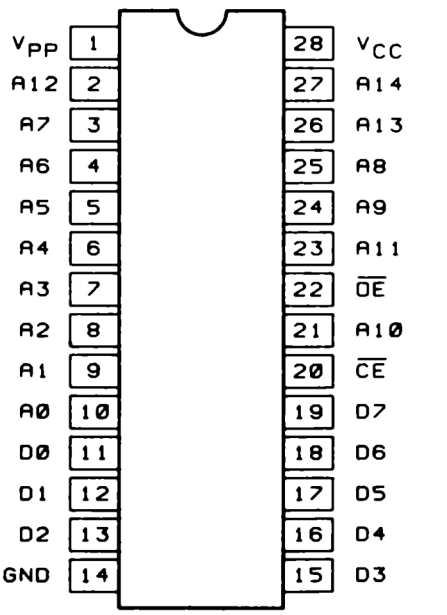
2564



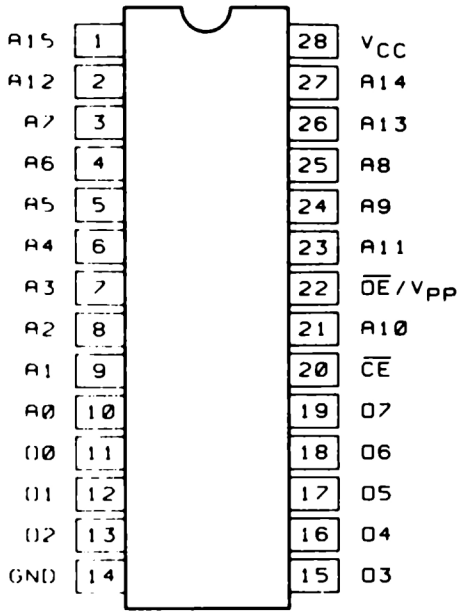
27128



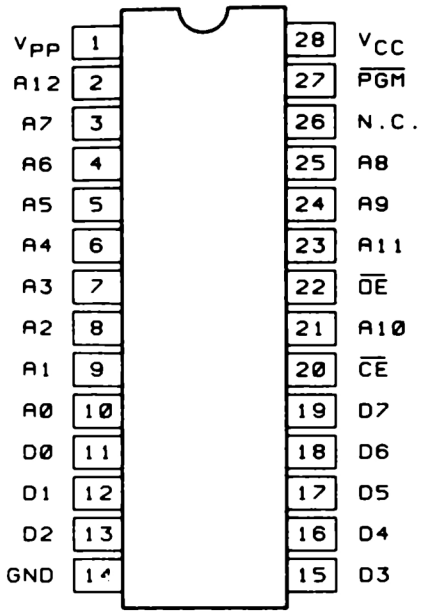
27256



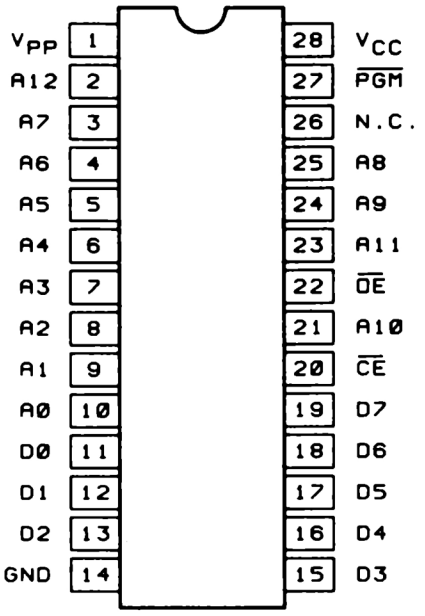
27512



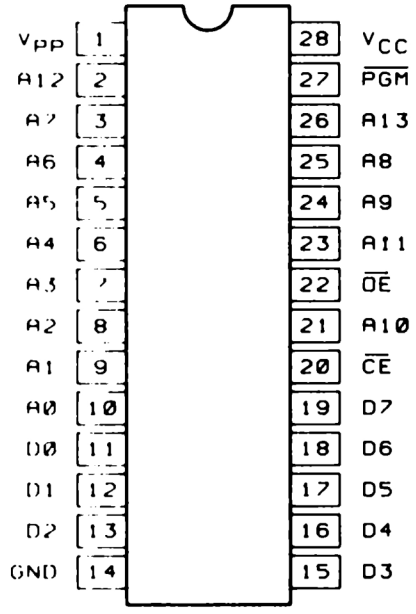
2764



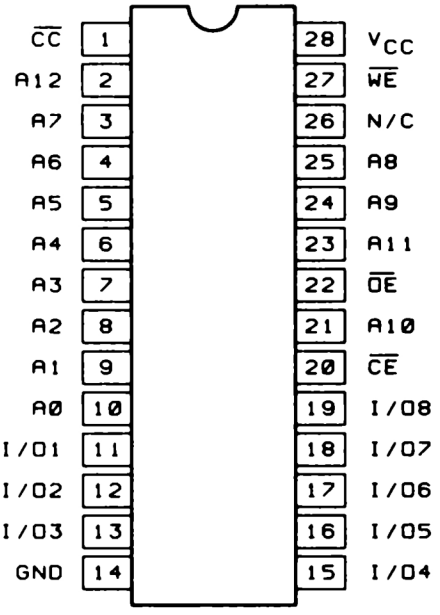
5133



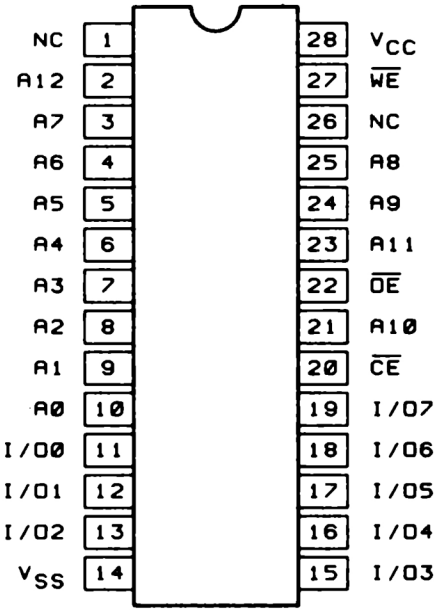
5143



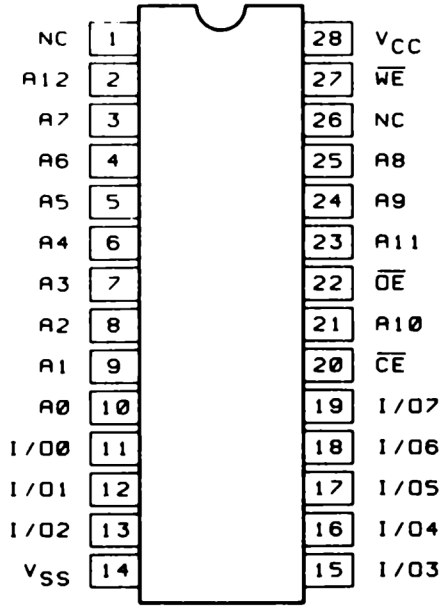
52B33



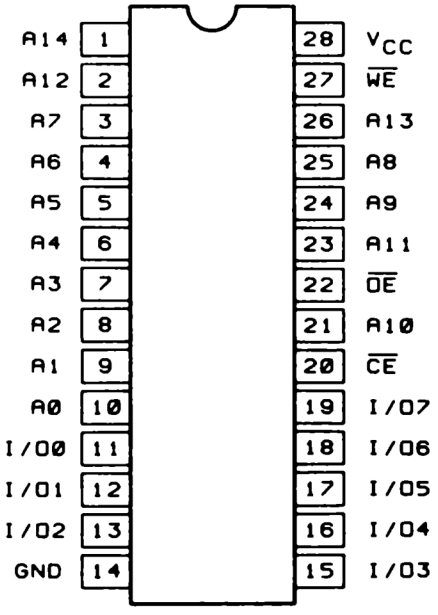
58064



X2864A



X28256A



Notizen:

MERLIN
DATA · ELEKTRONIK

PP-64 HANDBUCH - ZUSATZBESCHREIBUNG
* M O D U L G E N E R A T O R E N *

Generell ist bei Verwendung der Modulgeneratoren das Original C= 64 Betriebssystem zu benützen. Bei Verwendung anderer Betriebssysteme, ist eine einwandfreie Funktion der Modulgeneratoren nicht gewährleistet. Verwenden Sie zum generieren nur das Softwaremodul (Brennerplatine abstecken).

Wenn Sie im Arbeitsmenü den Menüpunkt "G" MODULGENERATOR anwählen, erscheint folgendes Untermenü:

```
MDE MODUL MAKER
===============

0...GENERATOR 16K
1...C= 64      64K
2...C128       64K
3...FREEZE FRAME
```

Mit der entsprechenden Ziffertaste (0, 1, 2 oder 3) können Sie hier nun den jeweiligen Modulgenerator anwählen.

0...GENERATOR 16K Siehe Handbuch Seite 7 "G MODULGENERATOR"

1...C=64 64K
Durch drücken der Taste 1 fragt Sie das Programm nach folgende Punkte:
STECKMODUL NAMEN (bei Verwendung des Moduls mit der MODULSTECKPLATZERWEITERUNG 4-fach wird im Menü beim jeweiligen Port dieser Name bis 16 Zeichen angezeigt). Geben Sie den Namen ein und/oder bestätigen Sie mit RETURN.
ANZAHL PROGRAMME: Mögliche Eingabe 1 bis 8. Sie können bis zu 8 Programme zu einem Modul zusammenfassen. Diese werden beim Autostart in Form eines Untermenüs mit den Ziffern 1 bis 8 senkrecht aufgelistet und durch drücken der jeweiligen Ziffer aufgerufen.
NAMEN IM MENÜ: Bei mehr als einem Programm können Sie nun mit einer Eingabe von 11 Zeichen das jeweilige Programm benennen. Nach Eingabe des letzten Namen wechselt der Bildschirm und es erscheint in der ersten Zeile

MODUL GENERIEREN: PROM: 8KB FREI:FE55

ANZAHL TEILE PRG. 1 _ (Zahl der Teile von Programm 1) 1 bis 8 Teile zulässig
FILENAME 1 _ Filenamen des 1. Teiles angeben
START & LÄNGE XXXX XXXX
STARTEN (J/N) _ bei Unterprogrammen "N"ein oder "J"a, jenachdem ob der Teil initialisiert werden muß oder nicht. Fragt das Programm nach einer Startadresse (bei Maschinenprogrammen), so muß diese in HEX angegeben werden.

WICHTIG! Bei mehrteiligen Programmen, muß das Hauptprogramm immer als letzter Teil generiert werden.

Dieser Vorgang wiederholt sich, um die bei der ANZAHL DER PROGRAMME angegebenen Zahl.

2...C128 64K

Dieser Modulgenerator ist für den C= 128 im 128'er Modus gemacht und generiert wie der Modulgenerator C= 64, durch drücken der Taste 2 fragt Sie das Programm nach den gleichen Punkte wie Punkt 1, zusätzlich werden Sie jedoch noch nach der STARTADRESSE und RAM BANK gefragt und diese müssen angegeben werden. Als Modulkarte dazu können Sie die EPROMKARTE 32/64 KB verwenden, welche nach einer kleinen Modifikation (eine Leiterbahn zwischen den beiden Lötaugen durchtrennen und entfernen einer Diode) im 128'er Modus benützbar ist.

3...FREEZE FRAME

Durch drücken der Taste 3 fragt Sie das Programm:

PROGRAMMNAME: _____ geben Sie hier den Namen des gefreezten Programmes vollständig ein (kein "*" benützen, da dieser Name als MODULNAME mit übernommen wird).

WICHTIG! Bei Benützung dieses Modulgenerators ist unbedingt zu beachten.
1.) Machen Sie bei FREEZE FRAME unbedingt den MEMORY RESET.
2.) Es darf nach dem FREEZER RESET nur mit "S" gesavt werden.
3.) Geben Sie zum Schluß den Namen so an, wie Sie diesen bei Verwendung mit der MODULSTECKPLATZERWEITERUNG 4-fach im Menü angezeigt haben möchten.
4.) Testen Sie zuerst das gefreezte Programm auf Funktionsfähigkeit.

!!! BESONDERS ZU BEACHTEN !!! ALLGEMEINE HINWEISE !!! BESONDERS ZU BEACHTEN !!!

Durch das Zusammenspiel MODULGENERATOR zur EPROMKARTE 32/64 KB werden bei Verwendung des Eproms 27512 das File "A.MODUL", welches die Modulgeneratoren 1 + 2 generieren und das File "A.NAME", welches der FREEZE FRAME Generator generiert in die 2. Hälfte des Eproms 27512 gebrannt (siehe PP-64 Handbuch Seite 3 "!!!Besonderheiten beim 27512!!!, Schalter links). Die jeweiligen 2. Files ("B.MODUL" und "B.NAME") in die ersten 32 KB des Eproms 27512 (Schalter rechts, oder bei alten Modellen Bügel zwischen Pin 1 + 14 einlegen).

Beim generieren von Modulen bis 32 KB erzeugen die Modulgeneratoren 1, 2 + 3 jeweils nur ein File (A.MODUL oder A.NAME).

Auf der EPROMKARTE 32/64 KB können Sie die beiden Sockel auch gemischt bestükken (1x Eprom 27256 und 1x Eprom 27512).

!!! BESONDERS ZU BEACHTEN !!! ALLGEMEINE HINWEISE !!! BESONDERS ZU BEACHTEN !!!

Kopieren Sie sich die Programme, welche Sie als Modul generieren möchten auf eine neu formatierte Diskette zusammen. Machen Sie sich vom Directory einen Ausdruck, damit Sie auch noch nachher genau wissen, wie die einzelnen Files heißen. Achten Sie darauf, daß 32 KB ca. 129 Blöck auf Diskette sind und 64 KB ca. 259 Blöcke. Zählen Sie vorher die Blöcke zusammen, damit Sie auch sicher die Programme als Modul generieren können.